On the l-Ary GCD-Algorithm in Rings of Integers

Douglas Wikström

Royal Institute of Technology (KTH)
KTH, Nada, S-100 44 Stockholm, Sweden

Abstract. We give an l-ary greatest common divisor algorithm in the ring of integers of any number field with class number 1, i.e., factorial rings of integers. The algorithm has a quadratic running time in the bit-size of the input using naive integer arithmetic.

1 Introduction

The greatest common divisor (GCD) of two integers a and b is the largest integer d such that d divides both a and b. The problem of finding the GCD of two integers efficiently is one of the oldest problems studied in number theory. The corresponding problem can be considered for two elements α and β in any factorial ring R. Then $\lambda \in R$ is a GCD of α and β if it divides both elements, and whenever $\lambda' \in R$ divides both α and β it also holds that λ' divides λ. A precise understanding of the complexity of different GCD algorithms gives a better understanding of the arithmetic in the domain under consideration.

1.1 Previous Work

The Euclidean GCD algorithm is well known. The basic idea of Euclid is that if $|a| \geq |b|$, then $|a \bmod b| < |b|$. Since we always have $\gcd(a, b) = \gcd(a \bmod b, b)$, this means that we can replace a with $a \bmod b$ without changing the GCD. Swapping the order of a and b does not change the GCD, so we can repeatedly reduce $|a|$ or $|b|$ until one becomes zero, at which point the other equals the GCD of the original inputs. In a more general setting with α and β in a factorial ring R, Euclid's idea works essentially unchanged if there exists a valuation $v : R \to \mathbb{R}^+$ with the following properties for $\alpha, \beta \in R$. There exists $\gamma, \delta \in R$ with $\alpha = \gamma\beta + \delta$ and $\delta = 0$ or $v(\delta) < v(\beta)$, if $\alpha\beta \neq 0$ then $v(\alpha) < v(\alpha\beta)$. Rings for which there exists such a valuation are called Euclidean. If in an algebraic ring $v(\alpha) = |N\alpha|$, where $N\alpha$ is the algebraic norm of α, the ring is called norm-Euclidean. Most algebraic rings are not even Euclidean. If we also want the Euclidean algorithm to terminate there must be a constant k such that $\{\alpha \mid v(\alpha) < k\}$ is finite.

All is however not lost. Kaltofen and Rolletschek [5] devise a GCD algorithm with quadratic running time for the ring of integers in any quadratic number field. Their approach is based on the idea to find an integer j such that $N(j\alpha \bmod$

L. Caires et al. (Eds.): ICALP 2005, LNCS 3580, pp. 1189–1201, 2005.
© Springer-Verlag Berlin Heidelberg 2005

$\beta) < N\beta$. This is always possible with $|j|$ bounded essentially by the square root of the discriminant. We are not aware of any generalization of this approach to more general rings of integers.

Interestingly, there are alternative approaches to compute the GCD. These are generalizations of Stein's binary GCD algorithm [9], which is particularly well suited for implementation on computers. It is based on the following facts.

$$\begin{array}{ll} \gcd(a,b) = 2\gcd(a/2,b/2) & \text{if } a \text{ and } b \text{ are even,} \\ \gcd(a,b) = \gcd(a/2,b) & \text{if } a \text{ is even and } b \text{ is odd, and} \\ \gcd(a,b) = \gcd((a-b)/2,b) & \text{if } a \text{ and } b \text{ are odd.} \end{array}$$

One may always apply one of the rules to reduce the size of elements, while preserving the GCD. Thus, by simply shifting and subtracting integers, the GCD of two integers can be computed. Weilert [11] generalizes this algorithm to the Gaussian integers. Damgård and Skovbjerg Frandsen [3, 4] independently also generalize the binary algorithm to the Eisenstein and Gaussian integers.

Sorenson [8] give the l-ary algorithm for computing the GCD of integers, which generalizes the binary algorithm. The l-ary algorithm is based on the result by Minkowski that given a and b one can find c and d such that $ca+db = 0 \bmod l$ for an integer l, where $|c|$ and $|d|$ essentially are bounded by $\sqrt{l}$. Thus, in each iteration the larger of a and b is replaced by $(ca + db)/l$. This is an analog to the binary algorithm, in that in each iteration the size of the largest integer is reduced roughly by a factor $2\sqrt{l}/l$. The details of this algorithm is slightly more involved than the binary algorithm, since the linear expression does not preserve the GCD. Sorenson also constructs a parallel version of his algorithm. Weilert [10] generalizes also this algorithm to the Gaussian integers.

Agarwal and Skovbjerg Frandsen [2] introduce an algorithm related to both the binary and the l-ary algorithms for computing GCD in several complex quadratic rings. It is interesting to note that one of the rings they consider is not Euclidean.

Wikström [13] generalizes the l-ary approach to compute the GCD in the ring of integers in the octic cyclotomic field. This is the first l-ary GCD algorithm in a non-quadratic ring, and the main inspiration to the current work.

The binary or l-ary GCD algorithm in the ring of integers $\mathbb{Z}[\zeta_m]$ of the cyclotomic number fields $\mathbb{Q}(\zeta_m)$ for $m = 2, 3, 4, 8$ can be "translated" to compute the corresponding power residue symbol. Shallit and Sorenson [7] give a binary algorithm for computing the Jacobi symbol. Weilert [11] generalizes the idea to compute the quartic residue symbol. Independently, both Damgård and Skovbjerg Frandsen [3, 4] and Wikström [12] generalize the idea to compute the cubic and quartic residue symbols. Wikström [13] also uses the idea to compute octic residue symbols.

1.2 Contribution

We give a GCD algorithm in the ring of integers $\mathcal{O}_K$ of any number field K with class number 1, i.e., rings of integers with unique factorization. Our result is non-uniform in the sense that we, for each ring, assume that we already know

an integral basis, fundamental units, and several constants derived from these. The running time of the algorithm is quadratic in the bit-size of the input.

As far as we know, the only previous generic GCD algorithm with quadratic running time is given by Kaltofen and Rolletschek [5], and this is only applicable to quadratic rings. The algorithm in [5] is in some sense "almost Euclidean", whereas our algorithm generalizes the l-ary algorithm [8]. As explained in the introduction, l-ary algorithms have appeared in the literature for specific rings, but the present work is the first to give a generic description of this approach.

We are confident that our algorithm can be "translated" to compute the m-th power residue symbol in the ring of integers of the m-th cyclotomic number field $\mathbb{Q}(\zeta_m)$ if it has class number 1.

Proofs of all claims are given in [14], but we try to convey the main ideas used in each proof here.

2 Notation

We denote the number of elements in a finite set A by $\#(A)$. We write $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, and $\mathbb{C}$ for the rational integers, the rational numbers, the real numbers, and the complex numbers. The imaginary unit is denoted by $i = \sqrt{-1}$. We denote the complex absolute value by $|\cdot| : \mathbb{C} \to \mathbb{R}$, where $|a + bi| = \sqrt{a^2 + b^2}$. We write $\overline{\alpha}$ to denote the complex conjugate of an $\alpha \in \mathbb{C}$.

Throughout the paper we use K to denote a number field with class number 1, and we use $\mathcal{O}_K$ to denote its ring of integers. Since $\mathbb{Q}$ is a perfect field $K/\mathbb{Q}$ is a separable extension. The ring $\mathcal{O}_K$ has an integral basis which we denote by $\omega_1, \ldots, \omega_g$, since $\mathcal{O}_K$ is the integral closure of $\mathbb{Z}$ which is a principal ideal domain. This means that $\mathcal{O}_K = \mathbb{Z}\omega_1 + \ldots + \mathbb{Z}\omega_g$ and $K = \mathbb{Q}\omega_1 + \ldots + \mathbb{Q}\omega_g$. We write $\mathcal{O}_K^*$ to denote the units of $\mathcal{O}_K$, i.e., the invertible elements. The corresponding notation is used also for other domains. We use $\varepsilon_1, \ldots, \varepsilon_h$ to denote a maximal set of independent fundamental units in $\mathcal{O}_K$. We denote the group of roots of unity by $\mu(K)$. We denote by $G = \mathrm{Hom}_{\mathbb{Q}}(K, \mathbb{C})$ the set of $\mathbb{Q}$-embeddings of K into $\mathbb{C}$, i.e., isomorphisms of K, which keep $\mathbb{Q}$ fixed. This implies that $g = \#(G)$. We assume throughout that $g \geq 2$. We use multiplicative notation for the action of an element $\sigma \in \mathrm{Hom}_{\mathbb{Q}}(K, \mathbb{C})$, i.e., $\sigma : \alpha \mapsto \alpha^\sigma$. We denote by $N\alpha = \prod_{\sigma \in G} \alpha^\sigma$ the algebraic norm of α. For $\alpha \in \mathcal{O}_K^*$, we have $N\alpha \in \mathbb{Z}$. We use the term irreducible only for non-units. One source for the above facts is Neukirch [6].

The naive complexity model we use in this paper stipulates that addition or subtraction of positive integers x and y takes time $O(\log x + \log y)$, and multiplication, integer division or computing remainders takes time $O(\log x \log y)$.

3 Preliminary Results

Before we describe the algorithm and analyze it we need to generalize the results given in [13].

3.1 Balanced Elements

Consider the absolute value of the algebraic norm, $|N\alpha| = \prod_{\sigma \in G} |\alpha^\sigma|$, of an element $\alpha \in \mathcal{O}_K$. It is given by the product of the absolute values of the conjugates of α. The quotient $|\alpha^\sigma|/|\alpha^{\sigma'}|$ of two such absolute values can be arbitrarily large for elements with a fixed absolute norm $|N\alpha|$. However, it follows from Dirichlet's Unit Theorem that there exists an associate β of α for which the absolute values $|\beta^\sigma|$ are roughly the same size. This is an important observation, since it allows us to establish a weak triangle inequality. Informally, we could say that we can balance the complex absolute values of the algebraic conjugates of α. We use the following definition.

Definition 1 (Δ-Balanced Element) *We say that a non-zero $\alpha \in K$ is Δ-*balanced *if $|\alpha^\sigma| \leq \Delta |\alpha^{\sigma'}|$ for all $\sigma, \sigma' \in \mathrm{Hom}_{\mathbb{Q}}(K, \mathbb{C})$.*

Note that α is Δ-balanced precisely when all of its conjugates are Δ-balanced, and that the requirement is equivalent to $\frac{1}{\Delta}|\alpha^\sigma| \leq |\alpha^{\sigma'}|$ for all $\sigma, \sigma' \in G$.

3.2 A Weak Triangle Inequality

It would be nice if given $\alpha, \beta \in K$, we had $|N(\alpha + \beta)| \leq c \max\{|N\alpha|, |N\beta|\}$ for a constant $c \in \mathbb{R}$, i.e., some type of "triangle inequality". Unfortunately, for almost all K there is no such law. Instead we show that there exists a triangle inequality for balanced elements.

Theorem 1 (Triangle Inequality for Δ-Balanced Elements) *Let α and β be Δ-balanced elements in K, and set $g = \#(\mathrm{Hom}_{\mathbb{Q}}(K, \mathbb{C}))$. Then*

$$|N(\alpha + \beta)| \leq 2^g \Delta^{g-1} \max\{|N\alpha|, |N\beta|\} \ .$$

The idea of the proof is to expand the product $|N(\alpha+\beta)| = |\prod_{\sigma \in G}(\alpha^\sigma + \beta^\sigma)|$ as a sum, apply the triangle inequality for the complex absolute value, and bound each term using the fact that α and β are balanced.

Remark 1 If the conjugates of α can be organized in pairs of complex conjugates one can give a slightly tighter inequality as is done in [13].

3.3 Linear Combinations

In this section we construct the cofactors of the l-ary approach, but first we exhibit a large set of elements with relatively small norm. Let $l \in \mathbb{Z}$, $l > 0$, denote a constant to be determined later and define the set

$$S_l = \left\{ \sum_{j=1}^{g} a_j \omega_j \;\middle|\; 0 \leq a_j \leq \sqrt{l} + 1 \right\} \ .$$

Each $\sigma \in G$ may be described as a $\mathbb{Z}$-linear map in the basis $\omega_1, \ldots, \omega_g$. We denote the matrix corresponding to this map by $f_\sigma = (f^\sigma_{k,j})_{1 \leq k,j \leq g}$, and define the constant $c_\omega = \max_{1 \leq k,j \leq g, \sigma \in G}\{|f^\sigma_{k,j}\omega_k|\}$. We have the following result.

Lemma 1 *Let $\gamma, \gamma' \in S_l$. Then for all $\sigma \in G$*

$$|(\gamma - \gamma')^\sigma| \leq g^2 c_\omega(\sqrt{l}+1) \ , \quad \textit{and} \quad \#(S_l) > l^{g/2} \ .$$

The first part of the lemma follows by the linearity of σ, application of the triangle inequality, and the fact that elements in S_l have small positive coefficients. The second part follows by counting.

Denote by T_l the set of pairwise differences $T_l = \{\gamma - \gamma' \mid \gamma, \gamma' \in S_l\}$. We show that for any Δ-balanced elements $\alpha, \beta \in \mathcal{O}_K$, we can find elements $\gamma, \delta \in T_l$ such that $l \mid (\gamma\alpha + \delta\beta)$ and still keep $|N(\gamma\alpha + \delta\beta)|$ relatively small. More precisely we define $C_{\text{lin}}(l) = (g^2 c_\omega(\sqrt{l}+1))^g \Delta^{g-1}$ and have the following theorem.

Theorem 2 *Let α and β be Δ-balanced elements in $\mathcal{O}_K$. Then there exists $\gamma, \delta \in T_l$, with $(\gamma, \delta) \neq (0,0)$, such that $l \mid (\gamma\alpha + \delta\beta)$, and*

$$|N(\gamma\alpha + \delta\beta)| \leq C_{\text{lin}}(l) \max\{|N\alpha|, |N\beta|\} \ .$$

The idea of the proof is as follows. The existence of the γ and δ follows by the pigeon-hole principle. The bound follows by an argument similar to that in the proof of Theorem 1, except that we apply Lemma 1 to bound the norm of the cofactors γ and δ.

In the following we need a notation to identify the cofactors guaranteed to exist by the theorem. We write $\gamma_{\alpha,\beta}$ and $\delta_{\alpha,\beta}$ for a pair of cofactors in T_l such that $l \mid (\gamma_{\alpha,\beta}\alpha + \delta_{\alpha,\beta}\beta)$.

3.4 Spurious Factors

Sorenson [8] notes that $\gcd(a,b) = \gcd(ca + db, b)$ may not hold for rational integers $a, b, c, d \in \mathbb{Z}$. A similar problem arises for algebraic integers. Fortunately, the following straightforward lemma explains this completely.

Lemma 2 *Let α, β, γ, and δ lie in $\mathcal{O}_K$. Then* $\gcd(\alpha, \beta) \mid \gcd(\gamma\alpha + \delta\beta, \beta)$ *and* $(\gcd(\gamma\alpha + \delta\beta, \beta) / \gcd(\alpha, \beta)) \mid \gamma$.

3.5 Approximating the Norm of a Δ-Balanced Element

The norm of an element gives in some sense the "size" of the element. Unfortunately, the way the norm is defined requires multiplication of integers, which takes time $O(n^2)$ in the naive arithmetic model. This is far too expensive to be done in each iteration of our algorithm, since we are looking for an algorithm that has a total running time of $O(n^2)$. It is natural to try to approximate the norm, but since elements can have small norm but large representation, i.e., be unbalanced, there may be much cancellation during the computation of the norm.

We consider a weaker estimate of the size of an element, which we call $N_+ : K \to \mathbb{R}$, and prove some useful results about this function. We do not know how to compute this function quickly, but in contrast to the norm it can be approximated within a constant factor in linear time for elements in $\mathcal{O}_K$.

Definition 2 *Define* $N_+ : K \to \mathbb{R}$ *by* $N_+\alpha = \max_{\sigma \in G}\{|\alpha^\sigma|\}$.

It is not hard to see that N_+ approximates the norm N arbitrarily badly, but it turns out to be useful anyway. The next lemma says that if an element is Δ-balanced, then N_+ is essentially a good approximation of the norm N.

Lemma 3 *Let* $\alpha \in K$ *be* Δ*-balanced. Then* $\sqrt[g]{|N\alpha|} \leq N_+\alpha \leq \Delta \sqrt[g]{|N\alpha|}$.

For the lemma to be useful there must be a way to balance an element α without computing its norm $N\alpha$, but we ignore this issue for now. Instead we introduce a function N'_+ which approximates N_+ within a constant factor.

Definition 3 *Define* $N'_+ : K \to \mathbb{R}$ *by* $N'_+\alpha = \max_{1 \leq j \leq g}\{|a_j|\}$ *for an element* $\alpha \in K$ *given by* $\alpha = \sum_{j=1}^{g} a_j\omega_j$ *with* $a_j \in \mathbb{Q}$.

The function N'_+ can obviously be evaluated in linear time in the bit-size of the input when $\alpha \in \mathcal{O}_K$, since then $a_j \in \mathbb{Z}$. Next we show that it approximates N_+ within a constant factor.

Denote by $K^*_{\mathbb{C}}$ the direct product $\prod_{\sigma \in G} \mathbb{C}^*$, and denote by $\psi : K \to K^*_{\mathbb{C}}$ the map given by $\psi : \alpha \mapsto (\alpha^\sigma)_{\sigma \in G}$. We consider K as a g-dimensional $\mathbb{Q}$-vector space, where elements are represented in the basis $\omega_1, \ldots, \omega_g$. Then the image $\psi(K)$ is an isomorphic $\mathbb{Q}$-vector space from the $\mathbb{Q}$-linearity of the homomorphisms. Denote by $(\psi_{\sigma,j})_{\sigma \in G, 1 \leq j \leq g}$ the complex valued matrix which represents the map $\psi : K \to \psi(K)$ expressed in the basis $\omega_1, \ldots, \omega_g$. Denote by $(\psi'_{j,\sigma})_{\sigma \in G, 1 \leq j \leq g}$ the complex valued matrices corresponding to the map $\psi^{-1} : \psi(K) \to K$ expressed in the canonical orthonormal basis $\{e_\sigma\}_{\sigma \in G}$ for $K_{\mathbb{C}} = \prod_{\sigma \in G} \mathbb{C}$. Define $\Gamma = g \max_{1 \leq j \leq g, \sigma \in G}\{|\psi_{\sigma,j}|, |\psi'_{j,\sigma}|\}$. The lemma below follows straightforwardly from the linearity of ψ and its inverse.

Lemma 4 *Let* $\alpha \in K$. *Then* $\frac{1}{\Gamma} N_+\alpha \leq N'_+\alpha \leq \Gamma N_+\alpha$.

Corollary 1 *Let* $\alpha \in K$ *be* Δ*-balanced. Then*

$$\frac{1}{\Gamma} \sqrt[g]{|N\alpha|} \leq N'_+\alpha \leq \Gamma\Delta \sqrt[g]{|N\alpha|} \ .$$

3.6 Balancing Elements

In this section we prove a result that allows us to balance elements in $\mathcal{O}_K$ efficiently. Recall the statement of Dirichlet's Unit Theorem. It considers a number field K which has r real embeddings and s pairs of conjugates of complex embeddings of K in $\mathbb{C}$, and says that the group of units $\mathcal{O}^*_K$ is the direct product of the group of roots of unity, $\mu(K)$, and a free abelian group of rank $r + s - 1$. The theorem itself is not strong enough for our purposes, but we can extract a useful result from the construction used in its proof. We follow the exposition given in Neukirch [6], but use slightly different notation.

We have already defined the map $\psi : K \to K^*_{\mathbb{C}}$. Denote by $\mathrm{vlog} : K^*_{\mathbb{C}} \to \prod_{\sigma \in G} \mathbb{R}$ the map given by $\mathrm{vlog} : (z_\sigma)_{\sigma \in G} \mapsto (\log |z_\sigma|)_{\sigma \in G}$. Conjugation $F : z \mapsto \overline{z}$ in $\mathbb{C}$ induces involutions. In $K^*_{\mathbb{C}}$ it acts by $F(z_\sigma)_{\sigma \in G} = (\overline{z_{\overline{\sigma}}})_{\sigma \in G}$ and in $\prod_{\sigma \in G} \mathbb{R}$

it acts by $F(x_\sigma)_{\sigma\in G} = (x_{\overline{\sigma}})_{\sigma\in G}$. We define $K_{\mathbb{R}}^*$ and $\left[\prod_{\sigma\in G}\mathbb{R}\right]^+$ to be the vector spaces consisting of fixed points of F in $K_{\mathbb{C}}^*$ and $\prod_{\sigma\in G}\mathbb{R}$ respectively. If σ is a real embedding, it is clearly fixed by F, and the complex embeddings comes in pairs. Since there are s pairs of complex embeddings we see that $\left[\prod_{\sigma\in G}\mathbb{R}\right]^+$ is isomorphic to $\mathbb{R}^{r+s}$.

Define $N_{\mathbb{C}} : K_{\mathbb{C}} \to \mathbb{C}$, $N_{\mathbb{C}} : (z_\sigma)_{\sigma\in G} \mapsto \prod_{\sigma\in G} z_\sigma$ and $\mathrm{Tr}_{\mathbb{R}} : \prod_{\sigma\in G}\mathbb{R} \to \mathbb{R}$, $\mathrm{Tr}_{\mathbb{R}} : (x_\sigma)_{\sigma\in G} \mapsto \sum_{\sigma\in G} x_\sigma$. It is shown in [6] that the following diagram commutes.

$$\begin{array}{ccccc} K^* & \xrightarrow{\psi} & K_{\mathbb{R}}^* & \xrightarrow{\mathrm{vlog}} & \left[\prod_{\sigma\in G}\mathbb{R}\right]^+ \\ \downarrow{\scriptstyle N} & & \downarrow{\scriptstyle N_{\mathbb{C}}} & & \downarrow{\scriptstyle \mathrm{Tr}_{\mathbb{R}}} \\ \mathbb{Q}^* & \longrightarrow & \mathbb{R}^* & \xrightarrow{\log} & \mathbb{R} \end{array}$$

Consider the following subgroups.

$$\begin{array}{rll} \mathcal{O}_K^* &= \{\varepsilon \in \mathcal{O}_K \mid N\varepsilon = \pm 1\} & \text{the units,} \\ S &= \{y \in K_{\mathbb{R}}^* \mid N_{\mathbb{C}} y = \pm 1\} & \text{the norm one surface, and} \\ H &= \{x \in \left[\prod_{\sigma\in G}\mathbb{R}\right]^+ \mid \mathrm{Tr}_{\mathbb{R}}(x) = 0\} & \text{the trace zero hyperplane.} \end{array}$$

The commutative diagram above induces the homomorphisms

$$\mathcal{O}_K^* \xrightarrow{\psi} S \xrightarrow{\mathrm{vlog}} H \ .$$

Denote by λ the composed map $\lambda = \mathrm{vlog} \circ \psi$, and let $L = \lambda(\mathcal{O}_K^*)$ be the image of the units in $\left[\prod_{\sigma\in G}\mathbb{R}\right]^+$. Recall the definition of a lattice.

Definition 4 *A lattice in an $\mathbb{R}$-vector space V is a subgroup $L = E_1\mathbb{Z} + \ldots + E_h\mathbb{Z}$, where $E_1, \ldots, E_h$ are linearly independent vectors in V. It is called complete if $E_1, \ldots, E_h$ is a basis for V.*

It is proved in [6] that the group of roots of unity, $\mu(K)$, is isomorphic to the kernel of λ and that L is a complete lattice in the $(r+s-1)$-dimensional vector space H. Dirichlet's theorem follows from this. Let $h = r + s - 1$. To define the fundamental units $\varepsilon_1, \ldots, \varepsilon_h$ we pick a basis $E_1, \ldots, E_h$ for L, and define $\varepsilon_j = \lambda^{-1}(E_j)$. We also define $E_j' = gE_j$.

Suppose we map an element α into $\left[\prod_{\sigma\in G}\mathbb{R}\right]^+$ using λ. Let $x = (x_\sigma) = \lambda(\alpha)$. Then it is not hard to see that α is balanced when all x_σ are of roughly the same size. Another way to phrase this is that the orthogonal projection of x onto H is close to the origin. If we multiply α by ε_j, the image $\lambda(\alpha\epsilon_j)$ is translated by the vector E_j, i.e., $\lambda(\alpha\varepsilon_j) = \lambda(\alpha) + \lambda(\varepsilon_j) = x + E_j$. To balance an element we want to find some integer combination of the vectors $E_1, \ldots, E_h$ that translates x close to the origin, since this corresponds to multiplying α by the fundamental units $\varepsilon_1, \ldots, \varepsilon_h$. We can always write $x = \sum_{j=1}^h r_j E_j$, with $r_j \in \mathbb{R}$, since L is a complete lattice, i.e., $E_1, \ldots, E_h$ is a $\mathbb{R}$-basis for H. Then we pick integers close to these real coefficients. Below we prove two lemmas that allow us to give a simple algorithm for balancing elements that is easy to analyze.

Recall that $\{e_\sigma\}_{\sigma \in G}$ denotes the canonical orthonormal basis for the space $\prod_{\sigma \in G} \mathbb{R}$. We define the max-norm $\| \cdot \| : H \rightarrow \mathbb{R}$ in terms of this basis by $\| \sum_{\sigma \in G} x_\sigma e_\sigma \| = \max_{\sigma \in G} \{|x_\sigma|\}$. It is intuitively clear that if an element $x \in H$ is sufficiently far from the origin, we may reduce its max-norm $\|x\|$ by an additive term t by translating it by a bounded element in the lattice L. We define a constant $c_E = \max_{r_1,\ldots,r_h \in [-1/2,1/2]} \| \sum_{j=1}^{h} r_j E'_j \|$, and turn this into a precise statement as follows. Denote by $A(t)$ the set $\{x \in H \mid x = \sum_{j=1}^{h} r_j E'_j, r_j \in \mathbb{R}, \|x\| \leq t + c_E\}$, and define

$$w(t) = \frac{1}{2} + \max_{\sum_{j=1}^{h} r_j E'_j \in A(t)} \{|r_j|\} \ ,$$

where $r_j \in \mathbb{R}$. We prove the following result.

Lemma 5 *Let $t > 0$ and let $x \in H$ be an element such that $\|x\| > t + c_E$. Then there exists $k_1, \ldots, k_h \in \mathbb{Z}$ with $|k_j| \leq w(t)$ such that*

$$\left\| x + \sum_{j=1}^{h} k_j E'_j \right\| \leq \|x\| - t \ .$$

Choose t such that $\frac{\Gamma^2}{2^t} < \frac{1}{2}$ and define the constants $\Phi = w(gt)$ and $\Delta = 2^{\frac{2}{g}(gt + c_E)}$. We translate the above lemma from the space H back to $\mathcal{O}_K$ and take care of the lack of precision in our approximation of N_+. This gives the following lemma.

Lemma 6 *If α in $\mathcal{O}_K$ is not Δ-balanced, then there exists $k_1, \ldots, k_h \in \mathbb{Z}$ with $|k_j| \leq \Phi$ such that*

$$N'_+ \left(\alpha \prod_{j=1}^{h} \varepsilon_j^{k_j} \right) < \frac{1}{2} N'_+ \alpha \ .$$

The idea of the proof is the following. Suppose α is not balanced and consider the element $\beta = \alpha^g / N\alpha$. Note that β is "normalized" in the sense that $x = \lambda(\beta) \in H$ (we may have $\beta \notin \mathcal{O}_K$ though). Up to a constant factor, the element α is balanced if and only if β is balanced. This implies that $x \in H$ is far from the origin. We then apply Lemma 5 to translate x closer to the origin. Since we do this using the basis $E'_1, \ldots, E'_h$ this translates to multiplying β by a product of the fundamental units ϵ_j. When this is no longer possible, x is close to the origin, which implies that β, and thus α, are balanced.

4 The Algorithm

In this section we describe the algorithm. We divide it into subroutine calls to improve readability.

4.1 Subroutines

Consider the set of non-unit elements that divide some δ in the set of cofactors T_l. This set is clearly infinite, since each element in $\mathcal{O}_K$ has an infinite number of associates. This makes it natural to consider the following set instead

$$F_l = \{\pi \in \mathcal{O}_K \ : \ \pi \mid l \text{ or } \pi \mid \delta \in T_l, \text{ and } \pi \text{ is } \Delta\text{-balanced and irreducible}\} \ .$$

The set F_l is bounded and we denote its elements by $F_l = \{\pi_1, \ldots, \pi_{s_F}\}$. We write $F_l \nmid \alpha$ to denote the fact that $\pi \nmid \alpha$ for all $\pi \in F_l$. For clarity we state trial division as an algorithm below.

Algorithm 1 (Extract Small Factors)

SMALL(α)
`Input :` $\alpha \in \mathcal{O}_K$.
`Output :` $(\alpha', (k_1, \ldots, k_{s_F}))$, where $\alpha = \alpha' \prod_{j=1}^{s_F} \pi_j^{k_j}$ and $F_l \nmid \alpha'$.
The algorithm is the trivial one. Find α', and k_j by trial division.

Lemma 7 *Let $\alpha \in \mathcal{O}_K$ and suppose $(\alpha', (k_1, \ldots, k_{s_F})) = \text{SMALL}(\alpha)$. Then the running time of the* SMALL*-algorithm on input α is $O(n(1 + \log \frac{|N\alpha|}{|N\alpha'|}))$.*

Note that $\pi \mid \alpha$ if and only if $N\pi \mid \frac{N\pi}{\pi}\alpha$. Since $\frac{N\pi}{\pi} \in \mathcal{O}_K$ this reduces, in linear time, trial division in $\mathcal{O}_K$ to trial division in $\mathbb{Z}$.

Next we consider the problem of Δ-balancing elements. The algorithm below repeatedly applies Lemma 6 to find an increasingly balanced associate of the input. When this is no longer possible, we know that the current associate is Δ-balanced.

Algorithm 2 (Balance Element)

BALANCE(α)
`Input :` $\alpha \in \mathcal{O}_K$.
`Output :` a Δ-balanced associate β of α.
$\beta \leftarrow \alpha$
`Do`
  $\alpha \leftarrow \beta$
  `For` $(k_1, \ldots, k_h) \in [-\Phi, \Phi]^h$ `Do`
    `If` $N'_+\left(\beta \prod_{j=1}^{h} \varepsilon_j^{k_j}\right) < \frac{1}{2} N'_+ \beta$ `Then`
      $\beta \leftarrow \beta \prod_{j=1}^{h} \varepsilon_j^{k_j}$
    `End If`
  `End For`
`While` $N'_+\beta < N'_+\alpha$
`Return` α

Lemma 8 *The output of the* BALANCE*-algorithm is Δ-balanced, and the algorithm runs in time $O\big(n(1 + \log(\max_{\sigma,\sigma' \in G} \frac{|\alpha^\sigma|}{|\alpha^{\sigma'}|}))\big)$.*

Identify $\mathcal{O}_K/(l)$ with the set of representatives $\sum_{j=1}^{g} a_j\omega_j$, with $0 \leq a_j < l$. Then let $(\gamma_{\alpha',\beta'}, \delta_{\alpha',\beta'})_{\alpha',\beta' \in \mathcal{O}_K/(l)}$ be the table of elements from T_l guaranteed to exist by Theorem 2, i.e., elements such that $l \mid (\gamma_{\alpha',\beta'}\alpha' + \delta_{\alpha',\beta'}\beta')$. For clarity we state finding the cofactors as an algorithm.

Algorithm 3 (Find γ and δ)

GAMMADELTA(α, β)
`Input :` $\alpha, \beta \in \mathcal{O}_K$.
`Output :` $(\gamma, \delta) \in T_l^2$, such that $l \mid (\gamma\alpha + \delta\beta)$.
Compute $\alpha' = \alpha \bmod (l)$ and $\beta' = \beta \bmod (l)$. Then output $(\gamma_{\alpha',\beta'}, \delta_{\alpha',\beta'})$.

Lemma 9 *The algorithm is correct and runs in time $O(n)$.*

4.2 Greatest Common Divisor

Finally, we are ready to give the algorithm for computing a greatest common divisor of two elements α and β in $\mathcal{O}_K$. The special case where one of the inputs is zero is treated in the first two lines. Then we extract all small factors of both inputs and store these. This allows us to determine all small factors in a GCD. Then we make sure that β is balanced. Consider now the while-loop of the algorithm. In each iteration α is balanced. This ensures that when we compute $N'_+\alpha$ and $N'_+\beta$ the results are in fact approximations of $\sqrt[g]{|N\alpha|}$ and $\sqrt[g]{|N\beta|}$. This gives us a good idea of which of the elements is the larger. The if-statement swaps α and β such that the norm of α is larger or at least within a constant factor of the norm of β. Then a linear expression is formed using the special

Algorithm 4 (Greatest Common Divisor)

GCD(α, β)
`Input :` $\alpha, \beta \in \mathcal{O}_K$, with either α or β non-zero.
`Output :` The greatest common divisor of α and β.
`If` $\alpha = 0$ `Return` β
`If` $\beta = 0$ `Return` α
$(\alpha, (k_1, \ldots, k_{s_F})) \leftarrow$ SMALL(α)
$(\beta, (k'_1, \ldots, k'_{s_F})) \leftarrow$ SMALL(β)
$\beta \leftarrow$ BALANCE(β)
`While` $\alpha \neq 0$ `Do`
  $\alpha \leftarrow$ BALANCE(α)
  `If` $N'_+\alpha < N'_+\beta$ `Then`
    $(\alpha, \beta) \leftarrow (\beta, \alpha)$
  `End If`
  $(\gamma, \delta) \leftarrow$ GAMMADELTA(α, β)
  $(\alpha, \cdot) \leftarrow$ SMALL$((\gamma\alpha + \delta\beta)/l)$
`Done`
`Return` $\beta \prod_{j=1}^{s_F} \pi_j^{\min\{k_j, k'_j\}}$

cofactors of bounded norm, and the result is divided by l. This reduces the norm of α. During the iterations of the while-loop spurious factors from the set F_l may be introduced into the current β. These are removed in the subroutine call, and the output is formed in the obvious way.

In each iteration, α, perhaps after swapping with β, is replaced by the expression $(\gamma\alpha + \delta\beta)/l$. We must obviously choose l large enough such that $|N((\gamma\alpha + \delta\beta)/l)| < |N\alpha|$. But, we must also take into account the lack of exactness in the approximation N'_+ of the norm used when deciding which of α and β is the larger. For simplicity we choose l such that the norm of α is guaranteed to be reduced by a factor of two in each iteration. More precisely we choose l as the smallest integer that satisfies the inequality $C_{\text{lin}}(l)/l^g < 1/(2\Gamma^{2g}\Delta^g)$. We can choose l to satisfy this inequality since $C_{\text{lin}}(l) = O(l^{g/2})$.

5 Analysis

In this section we prove the correctness of the algorithm and bound its running time. To simplify the exposition we denote by α_j and β_j, and α'_j and β'_j the values of α and β before and after the if-statement in the jth iteration of the while-loop.

Lemma 10 *The jth iteration, $j > 1$, runs in time $O\big(n(1+\log \frac{|N\alpha_j|\cdot|N\beta_j|}{|N\alpha_{j+1}|\cdot|N\beta_{j+1}|})\big)$.*

To see why the lemma is true, note that from the triangle inequality of the complex absolute value follows that in each iteration, $\max\{|\alpha^\sigma_{j+1}|\}$ can only be a constant factor larger than $\max\{|(\alpha'_j)^\sigma|\}$. This means that if α_{j+1} is very unbalanced, then $|N\alpha_{j+1}|$ must also be much smaller than $|N\alpha'_j|$. The lemma then follows from Lemma 7 and Lemma 8.

Theorem 3 *Algorithm 4 computes the greatest common divisor of its inputs in time $O(n^2)$ in the bit-size n of its input using naive arithmetic in $\mathbb{Z}$.*

The proof of correctness is straightforward except from the handling of spurious factors. Since all small factors are removed from both inputs and stored before the while-loop, any small factors found in the while-loop can safely be discarded. By Lemma 2, replacing α by $(\gamma\alpha+\delta\beta)/l$ preserves the GCD up to small factors. Since all small factors are removed by the call to the SMALL-algorithm the GCD of α and β is preserved and the output of the algorithm is correct. The bound of the running time is explained as follows. We have chosen l such that the absolute norm of one of the elements is reduced at least by a factor $1/2$ in each iteration. Since the norm is an integer and the algorithm halts when $\alpha = 0$, the algorithm executes at most $d = O(n)$ iterations. The subroutine calls made outside of the while-loop can be done in time $O(n^2)$. The running time of each iteration is bounded in Lemma 10. Thus, it remains to argue that the combined execution time $\sum_{j=2}^{d} O(n \log \frac{|N\alpha_j|\cdot|N\beta_j|}{|N\alpha_{j+1}|\cdot|N\beta_{j+1}|})$ of all subroutine calls sum to $O(n^2)$, but this follows by calculation.

6 On the Existence of Practical Algorithms

In this paper we focus on conceptual simplicity, and not on minimizing the constants in the running time. In particular the three subroutines SMALL, BALANCE, and GAMMADELTA are trivial brute force algorithms. Thus, an interesting line of future research is to device more efficient specialized versions of these subroutines, e.g., it should be possible to use lattice reduction techniques to balance elements. Another line of research is to exploit specific properties of $\mathcal{O}_K$ when $\#(G)$ is relatively small. In addition to the complex rings mentioned in the introduction, this seems possible for some real quadratic rings [1].

Acknowledgments

I wish to thank my advisor Johan Håstad for excellent advise. Without his help I would still be struggling. I also thank Torsten Ekedahl who essentially played the role of an extra advisor during this work.

References

1. S. Agarwal, *Personal communication*, November, 2004.
2. S. Agarwal, G. Skovbjerg Frandsen, *Binary GCD Like Algorithms for Some Complex Quadratic Rings*, ANTS 2004, LNCS 3076, pp. 57-71, 2004.
3. I. Damgård, G. Skovbjerg Frandsen, *Efficient Algorithms for gcd and Cubic Residuosity in the Ring of Eisenstein Integers*, BRICS Technical Report, ISSN 0909-0878, BRICS RS 03-8, 2003.
4. I. Damgård, G. Skovbjerg Frandsen, *Efficient algorithms for GCD and cubic residuosity in the ring of Eisenstein integers*, Fundamentals of computation theory, LNCS 2751, pp. 109-117, 2003 (revised version to appear in Journal of Symbolic Computation).
5. E. Kaltofen, H. Rolletschek, *Computing greatest common divisors and factorizations in quadratic number fields*, Mathematics of Computation, 53(188):697-720, 1989.
6. J. Neukirch, *Algebraic Number Theory*, ISBN 3-540-65399-6, Springer-Verlag Berlin, 1999.
7. J. Shallit, J. Sorenson, *A binary algorithm for the Jacobi symbol*, ACM SIGSAM Bulletin, 27 (1), pp. 4-11, 1993.
8. J. Sorenson, *Two Fast GCD Algorithms*, Journal of Algorithms, 16(1):110-144, 1994.
9. J. Stein, *Computational problems associated with Racah algebra*, Journal of Computational Physics No. 1, pp. 397-405, 1969.
10. A. Weilert, *Asymptotically fast GCD computation in* $\mathbb{Z}[i]$, In Algorithmic number theory (Leiden, 2000), LNCS 1838, pp. 595-613, 2000.
11. A. Weilert, $(1+i)$*-ary GCD computation in* $\mathbb{Z}[i]$ *as an analogue to the binary GCD algorithm*, Journal of Symbolic Computation, 30(5):605-617, 2000.

12. D. Wikström, *On the Security of Mix-Nets and Related Problems*, Licentiate thesis, Nada, KTH, TRITA-NA-04-06, ISSN: 0348-2952, ISRN KTH/NA/R--04/06--SE, ISBN 91-7283-717-9, May, 2004.
13. D. Wikström, *On the l-Ary GCD-Algorithm and Computing Residue Symbols*, Technical Report, Nada, KTH, Royal Institute of Technology, TRITA-NA-04-39, ISSN: 0348-2952, ISRN KTH/NA/R--04/39--SE, November, 2004.
14. D. Wikström, *On the l-Ary GCD-Algorithm in Rings of Integers*, Technical Report, Nada, KTH, Royal Institute of Technology, TRITA-NA-05-15, ISSN: 0348-2952, ISRN KTH/NA/R--05/15--SE, April, 2005.